Homework 3

Deepika Dilip

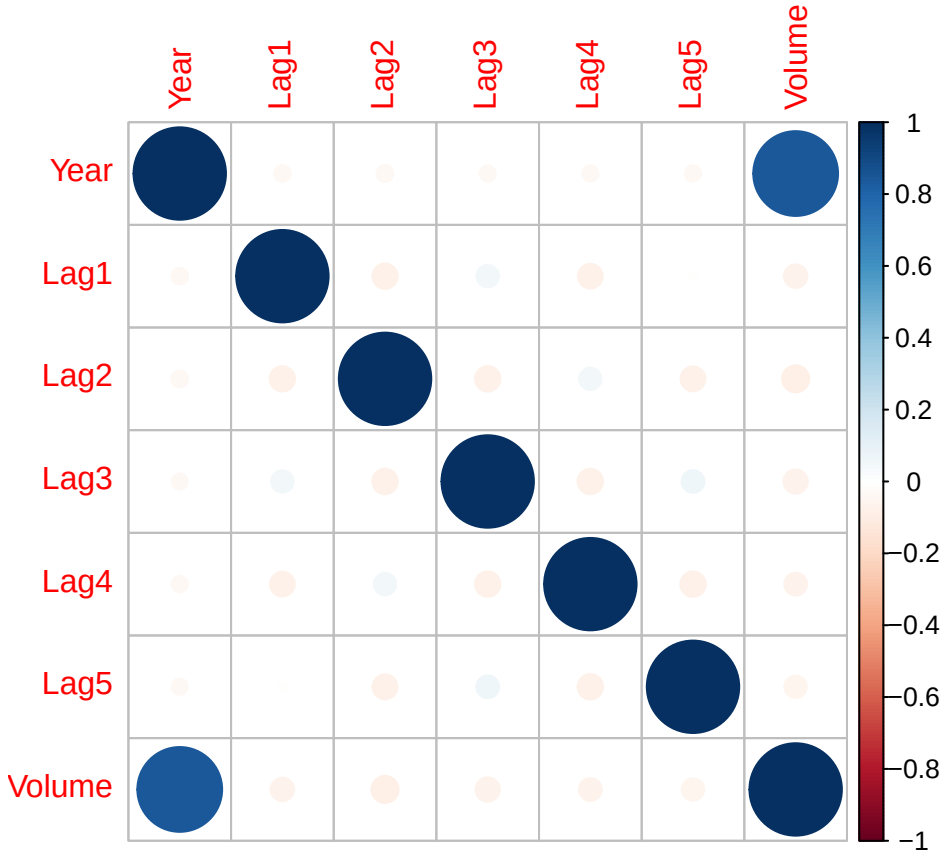
4/6/2019

Importing Data

```
Week1 <- Weekly[,-8]
```

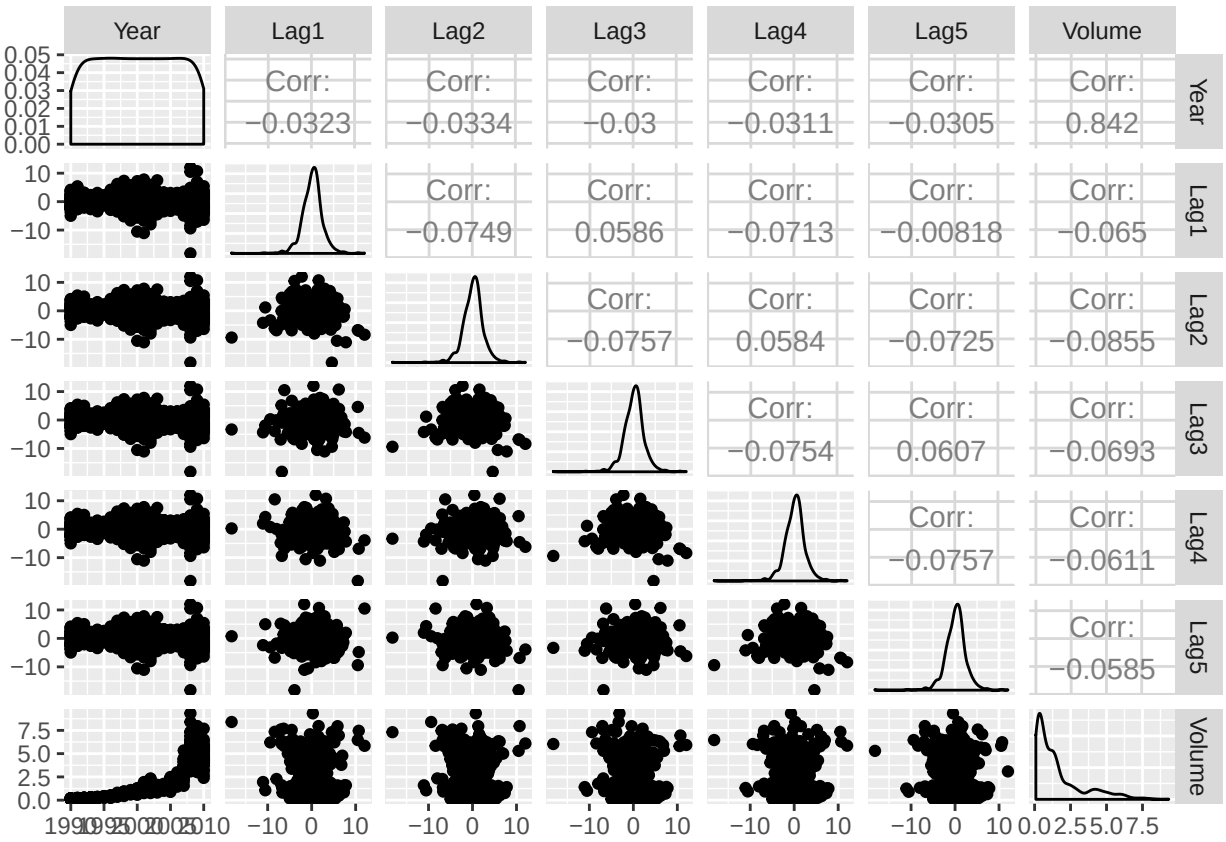
Part A: Graphical Summaries

```
x <- model.matrix(Direction~. ,Week1)[,-1]
y <- Week1$Direction
corrplot(cor(x))
```




```
y  <- as.numeric(y)
x <-as.data.frame(x)
```

```
ggpairs(x)
```



Here we have the correlation plot (showing correlation between predictors) and scatterplots between each predictor and outcome. We can see from the plots that with the exception of Volume and Year (that share a strong positive correlation), there doesn't seem to be too much correlation between predictors and the outcome.

Part B: Logistic Regression

```
set.seed(2)
log.fit <- glm(Direction ~ Volume + Lag1 + Lag2 + Lag3 + Lag4 + Lag5, family = "binomial", data=Week1)
tidy(log.fit)
```

```
## # A tibble: 7 x 5
##   term        estimate std.error statistic p.value
##   <chr>          <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)   0.267     0.0859     3.11  0.00190
## 2 Volume       -0.0227    0.0369    -0.616 0.538
## 3 Lag1         -0.0413    0.0264    -1.56  0.118
## 4 Lag2          0.0584    0.0269     2.18  0.0296
## 5 Lag3         -0.0161    0.0267    -0.602 0.547
## 6 Lag4         -0.0278    0.0265    -1.05  0.294
## 7 Lag5         -0.0145    0.0264    -0.549 0.583
```

At a 5% level of significance, the only predictor that is significant is Lag2 (Percentage return for 2 weeks previous).

Part C: Confusion Matrix

```
Direction = Week1$Direction
set.seed(2)
log.probs =predict(log.fit ,type = "response")
length(log.probs)
```

```
## [1] 1089
```

```
log.pred = rep("Down", 1089)
log.pred[log.probs >.5]="Up"
table(log.pred, Direction)
```

```
##         Direction
## log.pred Down  Up
##     Down   54  48
##     Up    430 557
```

```
mean(log.pred==Direction)
```

```
## [1] 0.5610652
```

56.10652% of predicted values are accurate. The confusion matrix gives us the information to calculate the sensitivity, specificity, PPV, and NPV. It helps produce values that assess how accurate our model is at predicting the outcome (in this case direction.)

Part D: ROC Curve

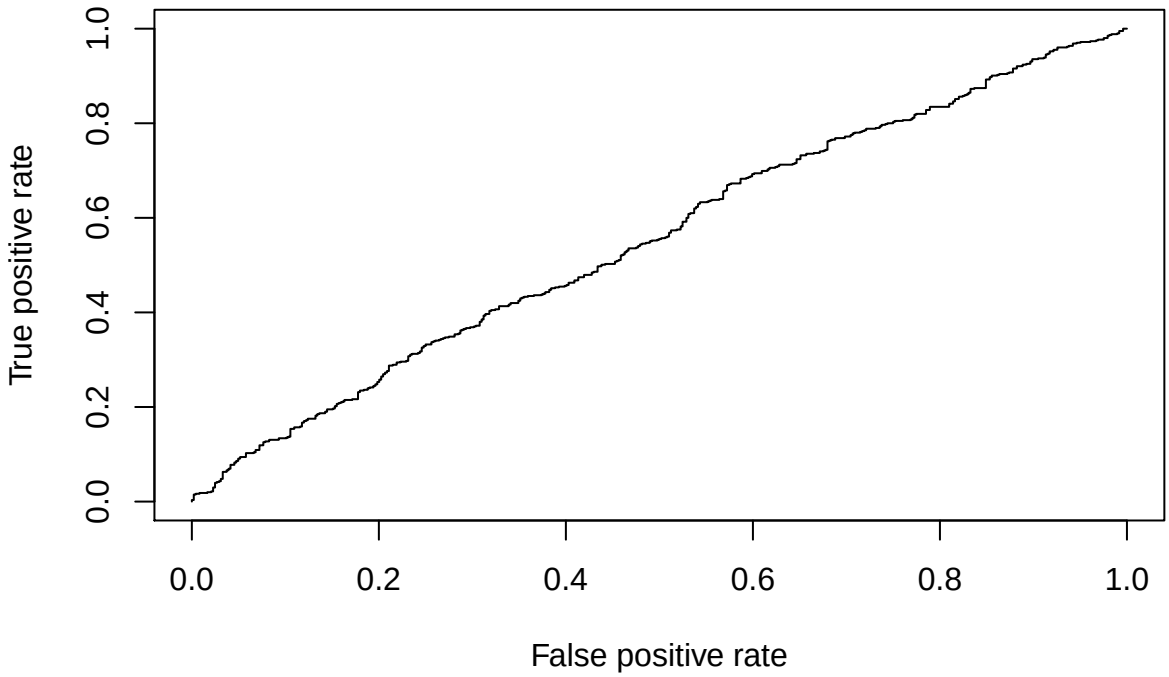
```
roc_table <-cbind(log.probs, Direction)
roc_table <- data.frame(roc_table)

#roc_table %>% mutate(log.pred = recode(log.pred, Up = "2")) %>% mutate(log.pred = recode(log.pred, Dow

roc_table$Direction <- as.numeric(roc_table$Direction)
roc_table$log.probs <- as.numeric(roc_table$log.probs)

pred <- prediction(as.numeric(roc_table$log.probs), as.numeric(roc_table$Direction))
```

```
roc.perf = performance(pred, measure = "tpr", x.measure = "fpr")
plot(roc.perf)
```




```
auc.perf = performance(pred, measure = "auc")
print(auc.perf@y.values)
```

```
## [[1]]
## [1] 0.5536985
```

```
#plot.roc(as.numeric(roc_table$log.pred), as.numeric(roc_table$Direction))
```

The Area Under the Curve is 0.5536985.

Part E: Logistic Model w/Training and Testing Sets

```
#Partitioning training and testing data
set.seed(2)
train <- Week1[ which(Week1$Year < 2009), ]
test <- Week1[ which(Week1$Year >= 2009), ]
log2.fit <- glm(Direction ~ Lag1 + Lag2, family = "binomial", data=train)
set.seed(2)

#Predicting probabilities
log2.probs = predict(log2.fit, test, type="response")

#Converting probabilities to predicted values
length(log2.probs)
```

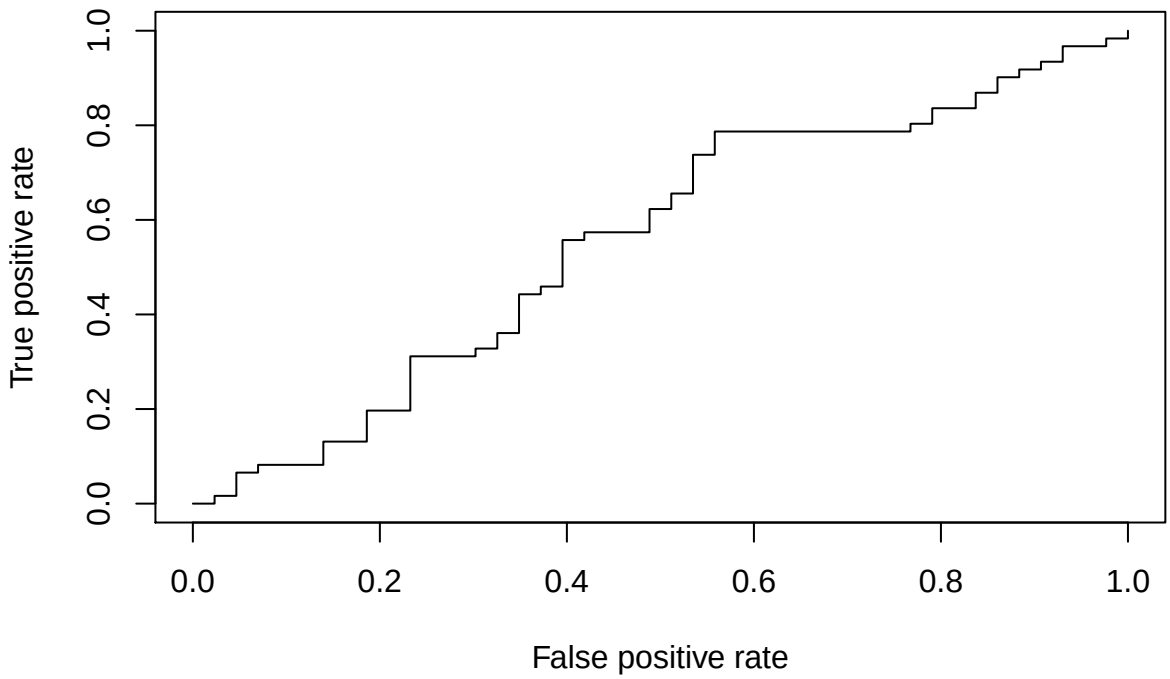
```
## [1] 104
```

```
log2.pred = rep("Down", 104)
log2.pred[log2.probs >.5]="Up"
log2.pred <- recode(log2.pred, Up = "2", Down = "1")

#Putting Predicted and Actual Values Together
roc_table2 <-cbind(log2.probs, test$Direction)
```

```
roc_table2 <- data.frame(roc_table2)

pred2 <- prediction(as.numeric(roc_table2$log2.probs), as.numeric(roc_table2$V2))
roc.perf2 = performance(pred2, measure = "tpr", x.measure = "fpr")
plot(roc.perf2)
```




```
auc.perf = performance(pred2, measure = "auc")
print(auc.perf@y.values)
```

```
## [[1]]
## [1] 0.5558521
```

AUC is 0.5558521

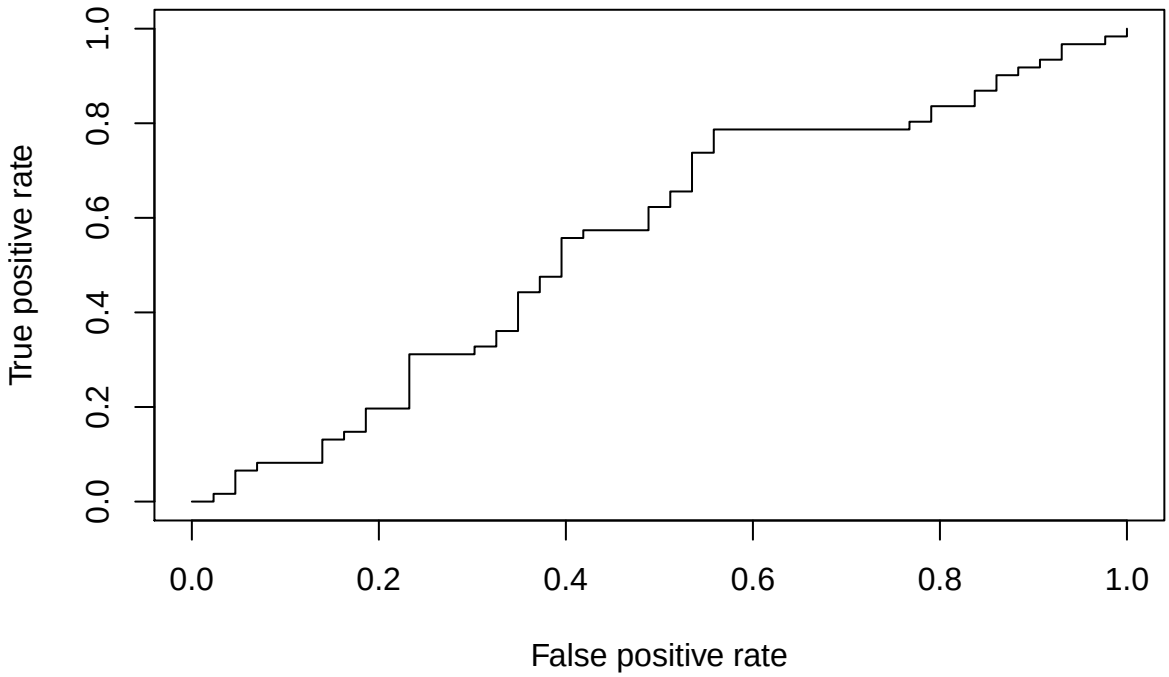
Part F: Linear Discriminant Analysis Model w/Training and Testing Sets

```
lda.fit <- lda (Direction ~ Lag1 + Lag2, data=train)
lda.pred=predict (lda.fit, test)

#Putting Predicted and Actual Probabilities Together
roc_table_lda <-cbind(lda.pred$posterior[,2], test$Direction)
roc_table_lda <- data.frame(roc_table_lda)

#ROC Curve
#lot.roc(as.numeric(roc_table_lda$X1), as.numeric(roc_table_lda$X2))

pred_lda <- prediction(as.numeric(roc_table_lda$X1), as.numeric(roc_table_lda$X2))
roc.perf.lda = performance(pred_lda, measure = "tpr", x.measure = "fpr")
plot(roc.perf.lda)
```




```
auc.perf.lda = performance(pred_lda, measure = "auc")
print(auc.perf.lda@y.values)
```

```
## [[1]]
## [1] 0.5566146
```

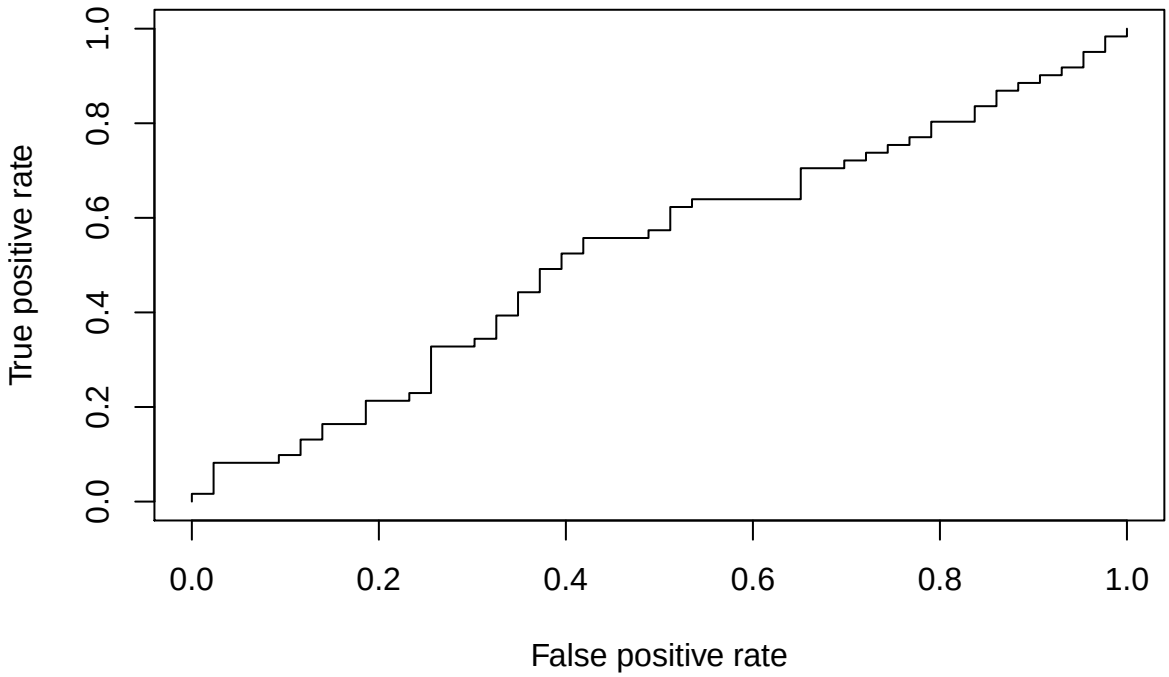
AUC is 0.5566146.

Part F: Quadratic Discriminant Analysis Model w/Training and Testing Sets

```
qda.fit <- qda (Direction ~ Lag1 + Lag2, data=train)
qda.pred=predict (qda.fit, test)

#Putting Probabilities and Actual Values Together
roc_table_qda <-cbind(qda.pred$posterior[,1], test$Direction)
roc_table_qda <- data.frame(roc_table_qda)

#ROC Curve
pred_qda <- prediction(as.numeric(roc_table_qda$X1), as.numeric(roc_table_qda$X2))
roc.perf.qda = performance(pred_qda, measure = "tpr", x.measure = "fpr")
plot(roc.perf.qda)
```




```
auc.perf.qda = performance(pred_qda, measure = "auc")
print(auc.perf.qda@y.values)
```

```
## [[1]]
## [1] 0.5287838
```

AUC is 0.5287838.

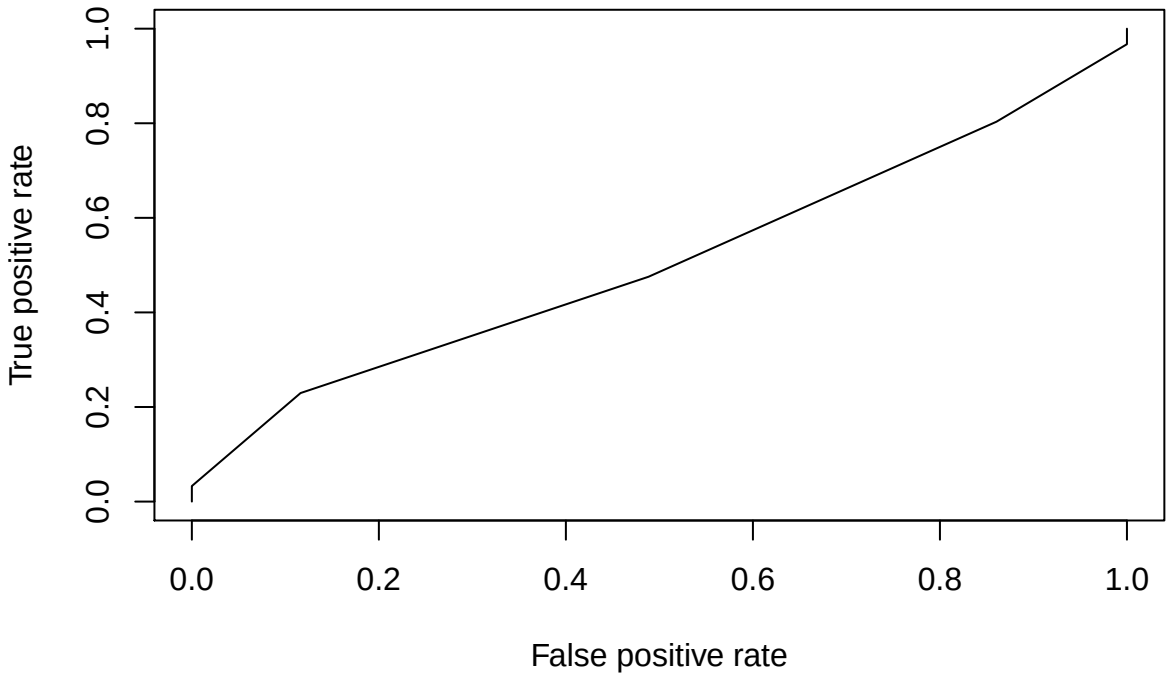
Part G: kNN Model w/Training and Testing Sets

```
trctrl <- trainControl(method = "repeatedcv", classProbs=T, number = 10, repeats = 5)
set.seed(2)
knn_fit <- train(Direction ~ Lag1 + Lag2, data = train, method = "knn",
 trControl=trctrl,
 preProcess = c("center", "scale"),
 tuneLength = 10, metric='ROC')

knn_prob <-predict(knn_fit, newdata = test, type="prob")[,2]


#Plotting ROC
pred_knn <- prediction(as.numeric(knn_prob), as.numeric(test$Direction))

roc.perf.knn = performance(pred_knn, measure = "tpr", x.measure = "fpr")
plot(roc.perf.knn)
```




```
auc.perf.knn = performance(pred_knn, measure = "auc")
print(auc.perf.knn@y.values)
```

```
## [[1]]
## [1] 0.5078155
```

AUC is 0.5078155.

Based on the results of kNN, it does not seem to be a more accurate model, as it has a lower AUC (hence lower classifier performance) compared to the other models ran. previously. This might be due to the nature of the data–in this case, classifier methods are more effective than non-parametric methods. Overall, the logistic regression method was the best–and even that one was slightly better than chance. If we could use predictive modeling on stocks, we probably would be doing better things than attending graduate school.